REMARKS

In the above-identified Office Action, the Examiner Claims 1 – 3, 6 – 8, 11 – 13 and 16 - 18 under 35 U.S.C. §102(b) as being anticipated by Sukegawa et al. Claims 4, 5, 9, 10, 14, 15, 19 and 20 were rejected under 35 U.S.C. §103(a) as being unpatentable Sukegawa et al. in view of Potter.

In reviewing the claims, Applicants noticed a minor error in Claim 16. Applicants have amended the claim to correct the error. For the reasons stated more fully below, Applicants submit that the claims are allowable over the applied references. Hence, reconsideration, allowance and passage to issue are respectfully requested.

As disclosed in the Response to the first Office Action as well as in the SPECIFICATION, multiprocessor systems are often connected to a network or networks through a limited number (usually one) of physical interfaces. Consequently, before a processor in a multiprocessor system uses a physical interface to transmit network data, it has to first request permission to lock out all the other processors from using the interface. If more than one processor is requesting access to the interface, there may be some access contention or lock contention. To reduce the likelihood of lock contention, an algorithm is generally used to select which one of the requests to honor first. The algorithm may do so on a first-come, first serve or round robin or on a priority basis or using any other contention resolution scheme.

When an access request is honored, the requesting processor is allowed to lock out all other processors from using the interface until the data is transmitted. When the processor has finished transmitting the data, it releases the lock to allow another processor to gain access to the lock. Obviously, while the processor is transmitting data, other processors may issue requests to the lock. Hence, there may be instances when other processors have to wait before gaining access to the physical interface in order to transmit data. In these

Instances, the physical interface may be viewed as a bottleneck as requests for the physical interface are accumulating at that point.

Thus, although the use of a multiprocessor in a system may greatly improve a computer system's performance, network communications performance may nonetheless not benefit from the use of the multiple processors due to this bottleneck. Therefore, it would be desirable to have a method that alleviates bottlenecks at the physical interface in the point of view of the processors. The present invention provides such a method.

According to the teachings of the invention, when a multiprocessor system that uses a limited number of physical interfaces is to transact data, a determination is made as to whether the data is network data. If the data is network data, the data is transmitted using a virtual Internet protocol (IP) address. The virtual IP address is the IP address of a data holding device rather than the address of a receiving computer.

Thus, the data before being sent onto the network to the receiving computer is sent to the data holding device. This frees up the processors of the multiprocessor system to continue to process data instead of becoming idle, waiting for the data to be transmitted. This may greatly enhance the performance of the multiprocessor system, especially in the case where there is a (long) queue to transmit data onto the network.

The invention is set forth in claims of varying scopes of which Claim 1 is illustrative.

> 1. A method of improving performance in a multiprocessor system that uses a limited number of physical interfaces to transact network data comprising the steps of:
>
> determining whether data being processed is network data; and
>
> ***transacting, if the data is network data, the data using a virtual Internet protocol (IP) address, the virtual IP address being an IP address given to a data***

> ***holding device in the multiprocessor system***. (Emphasis added.)

The Examiner rejected the independent claims under 35 U.S.C. §102(b) as being anticipated by Sukegawa et al. Applicants respectfully disagree.

Sukegawa et al. purport to teach a method of interprocessor data transfer using a network, virtual addresses and paging, a buffer, flags, data transfer status information and user accessible storage areas in main memory. To send interprocessor data with data receive area not fixed in the real memory and with less overhead for synchronization, the send node sends to the destination node, data, a virtual address of a receive area, an address of a receive control flag, a comparison value, and a comparison method. Network adaptor in the destination node judges whether the transfer condition is fulfilled, based on the comparison value, the comparison method and the semaphore in the receive control flag designated by the receive control flag address. Network adaptor further detects whether the receive area of the virtual address is in the main storage, based on the virtual address and an address translation table. The send data is stored in the receive buffer provided in the area for OS, when the above-mentioned condition is not fulfilled or the receive area is not in the main storage. Either when the destination node program issues a specific system call or when the program issues a reading instruction to the data in the receive area and a page fault is generated, OS moves the send data from the receive buffer to the receive area.

Thus, the virtual address advocated by Sukegawa et al. is memory address and not Internet Protocol (IP) address. It is well known that memory addresses and IP addresses are two different kinds of addresses. A memory address is an address of a location in a memory system of a computer where data is (to be) stored. By contrast, an IP address is an address of a computing device in a network.

Therefore, Sukegawa et al. do not teach the step of ***transacting, if the data is network data, the data using a virtual Internet protocol (IP) address***,

the virtual IP address being an IP address given to a data holding device in the multiprocessor system as claimed in Claim 1.

Consequently, Sukegawa et al. do not anticipate Claim 1. Hence, Applicants submit that Claim 1, as well as its dependent claims, are allowable over Sukegawa et al. Independent Claims 6, 11 and 16, which all incorporate the above-emboldened-italicized limitations in the above-reproduced claim 1, together with their dependent claims should also be allowable. Hence, Applicants once more respectfully request reconsideration, allowance and passage to issue of the claims in the application.

Respectfully Submitted

By: _____
Volel Emile
Attorney for Applicants
Registration No. 39,969
(512) 306-7969